

# 操作手順書

## データ解釈モジュール

V e r . 1 . 1

2 0 1 1 年 1 2 月 0 3 日

東京大学 浅間研究室

## 改版履歷

[illegible]

# 目次

改版履歴 .....	i
目次.....	ii
1. はじめに .....	1
1. 1. 本書の適用範囲.....	1
1. 2. 関連文書.....	1
1. 3. 本書を読むにあたって .....	1
1. 4. 動作環境.....	2
2. ソースディレクトリ構成 .....	3
3. ソフトウェアインストール.....	4
3. 1. 基本環境.....	4
3. 1. 1. OpenRTM-aist 1.0.0-RELEASE (Java 版) .....	4
3. 1. 2. Java 開発環境 .....	4
3. 1. 3. OpenRTM Eclipse tools 1.1.0-RC1.....	4
3. 1. 4. タウンマネジメントシステム RTC.....	4
3. 1. 5. データ解釈モジュール .....	5
4. 実行.....	6
4. 1. ネームサーバの起動.....	6
4. 2. RTSystemEditor の起動 .....	6
4. 3. diConsumer の起動 .....	6
4. 4. diProvider の起動.....	6
4. 5. TmsRtProvider の起動.....	6
4. 6. RTC のリンク .....	7
4. 7. 実行.....	8
4. 8. 終了手順.....	9
5. 特記事項 .....	10

# 1. はじめに

## 1. 1. 本書の適用範囲

本書はデータ解釈モジュールを、産業技術総合研究所が開発した RT ミドルウェア OpenRTM-aist を用いて動作させるための手順について記述する。

## 1. 2. 関連文書

本書の関連文書は下表の通り。

表 1-1 関連文書

No.	文書名	備考
1	データ解釈モジュール機能仕様書	データ解釈モジュールの仕様について記載

## 1. 3. 本書を読むにあたって

本書は RT ミドルウェア、RT コンポーネント（以下、RTC）、およびタウンマネジメントシステム（以下、TMS）に関する基礎知識を備えた利用者を対象としている。RT ミドルウェア、RTC、TMS については下記を参照のこと。

OpenRTM-aist Official Website:

<http://www.openrtm.org/>

タウンマネジメントシステム Website:

[http://www.openrtm.org/openrtm/ja/project/NEDO\\_Intelligent\\_PRJ\\_ID121](http://www.openrtm.org/openrtm/ja/project/NEDO_Intelligent_PRJ_ID121)

[http://210.154.184.16/pukiwiki/?ID\\_121](http://210.154.184.16/pukiwiki/?ID_121)

## 1. 4. 動作環境

検証に用いた動作環境は以下のとおりである。

OS	Microsoft Windows XP Professional SP3
RT ミドルウェア	OpenRTM-aist 1.0.0 RELEASE(Java)
開発言語	Java
コンパイラ	JDK1.6
依存ライブラリ (OpenRTM)	なし
依存ライブラリ (その他)	TMS Server
	TMS API

## 2. ソースディレクトリ構成

本書では下表のディレクトリ構成のもとに記述する。ディレクトリ構成が下表と異なる場合は、適宜その環境に合わせた修正が必要となる。

ディレクトリ	言語	内容	備考
C:\¥DataInterPretation	-	-	-
└ diProvider	-	データ解釈モジュール	-
└└ bin	-	-	-
└└ classes	-	実行バッチファイル	-
└└ com	-	-	-
└└ jp	-	-	-
└└ src	-	-	-
└ diConsumer	-	動作確認用コンポーネント	-
└└ bin	-	-	-
└└ classes	-	実行バッチファイル	-
└└ jp	-	-	-
└└ src	-	-	-

## 3. ソフトウェアインストール

### 3. 1. 基本環境

#### 3. 1. 1. OpenRTM-aist 1.0.0-RELEASE (Java 版)

Java 言語で実装された RTC を実行するための RT ミドルウェアをインストールする。インストールの詳細は、以下のサイトを参照のこと

ダウンロード先 URL: OpenRTM-aist-Java-1.0.0-RELEASE

<http://www.openrtm.org/openrtm/ja/node/933>

#### 3. 1. 2. Java 開発環境

OpenRTM-aist-Java-1.0.0 の動作および開発に必要な JDK をインストールする。

ダウンロード先 URL: JDK5.0: Java SE Development Kit Web ページ

<http://www.oracle.com/technetwork/java/javase/downloads/index-jdk5-jsp-142662.html>

#### 3. 1. 3. OpenRTM Eclipse tools 1.1.0-RC1

RT コンポーネントを作成したり、操作するためのツールをインストールする。インストールの詳細は、以下のサイトを参照のこと

ダウンロード先 URL: OpenRTM Eclipse tools 1.1.0-RC1

<http://www.openrtm.org/openrtm/ja/node/1736>

#### 3. 1. 4. タウンマネジメントシステム RTC

OpenRTM 上で、タウンマネジメントシステムの機能を利用するためのタウンマネジメントシステム RTC を以下の URL を通してダウンロードする。

ダウンロード先 URL: RTC 再利用 Web

[http://www.openrtp.jp/reuse\\_web/top.php](http://www.openrtp.jp/reuse_web/top.php)

ユーザーID, パスワードを入力しログインした後, モジュール一覧よりタウンマネジメントシステム RTC をダウンロードする。

また、タウンマネジメントシステム RTC の操作手順書の通りに TMS Sever、TMS API のインストール等の事前準備を行う。

### 3. 1. 5. データ解釈モジュール

データ解釈モジュールを、2章で示したディレクトリ構成と同じ構成となるように配置する。



## 4. 実行

本章では、データ解釈モジュールの実行手順を示す。なお、TMS Server は起動しているとする。

### 4. 1. ネームサーバの起動

OpenRTM-aist の”Start Naming Service”を実行してネームサーバを起動する。Windows XP の場合は、スタートメニューから “すべてのプログラム” → “OpenRTM-aist” → “C++” → “tools” → “Start Naming Service” より実行します。

### 4. 2. RTSystemEditor の起動

RT System Editor を起動する。Windows XP の場合は、スタートメニューから “すべてのプログラム” → “OpenRTM-aist” → “C++” → “tools” → “RT System Editor” より実行します。RT System Editor の Name Service View でネームサーバが起動していることを確認する。

### 4. 3. diConsumer の起動

以下のバッチファイルを実行して diConsumer を起動する。

```
C:\¥DataInterpretation¥diConsumer¥classes¥diConsumer.bat
```

起動後に RT System Editor の Name Service View に、diConsumer が登録されていることを確認する。

### 4. 4. diProvider の起動

以下のバッチファイルを実行して diProvider を起動する。

```
C:\¥DataInterpretation¥diProvider¥classes¥diProvider.bat
```

起動後に RT System Editor の Name Service View に、diProvider が登録されていることを確認する。

### 4. 5. TmsRtProvider の起動

TmsRtProvider の実行環境が格納されているディレクトリ内の TmsRtProviderComp.bat を実行する。詳細は、タウンマネジメント RTC の操作手順書を参照のこと。

起動後に RT System Editor の Name Service View に、TmsRtProvider が登録されていることを確認する。

## 4. 6. RTC のリンク

RT System Editor を用いて diConsumer、diProvider、TmsRtProvider を接続する。System Diagram に diConsumer、diProvider、TmsRtProvider をドラッグアンドドロップし、全てのサービスポートを接続する。接続が完了したら、ツールバーの “All Activate” をクリックすることにより活性化を行う。接続および活性化が行われた状態を下図に示す。

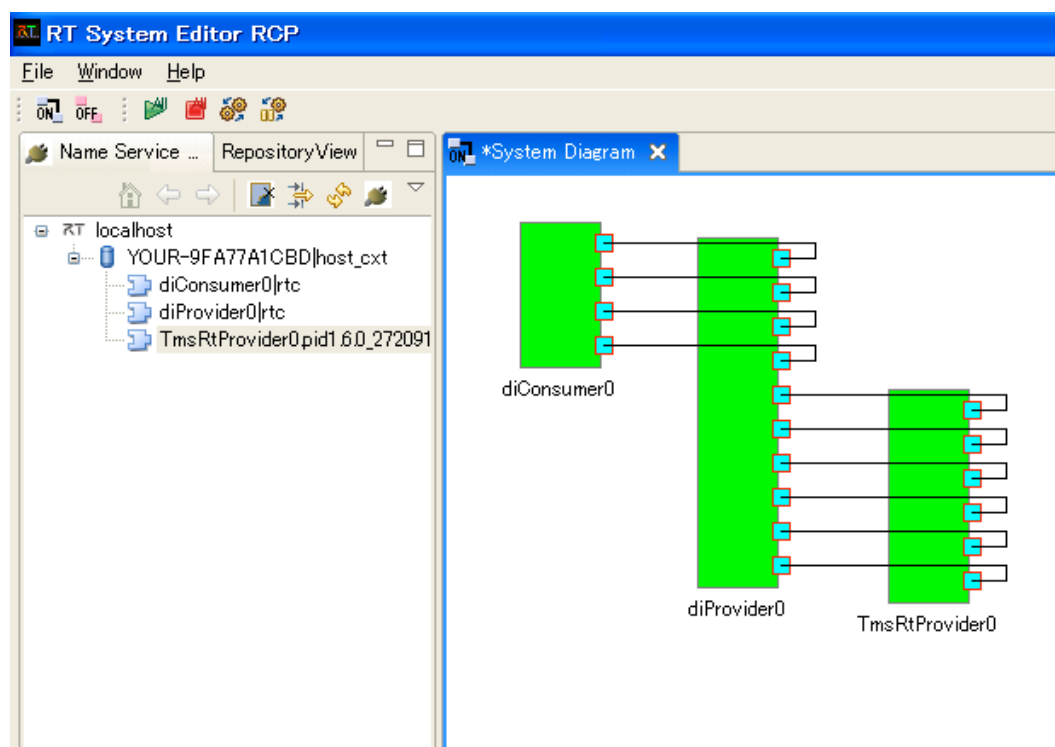
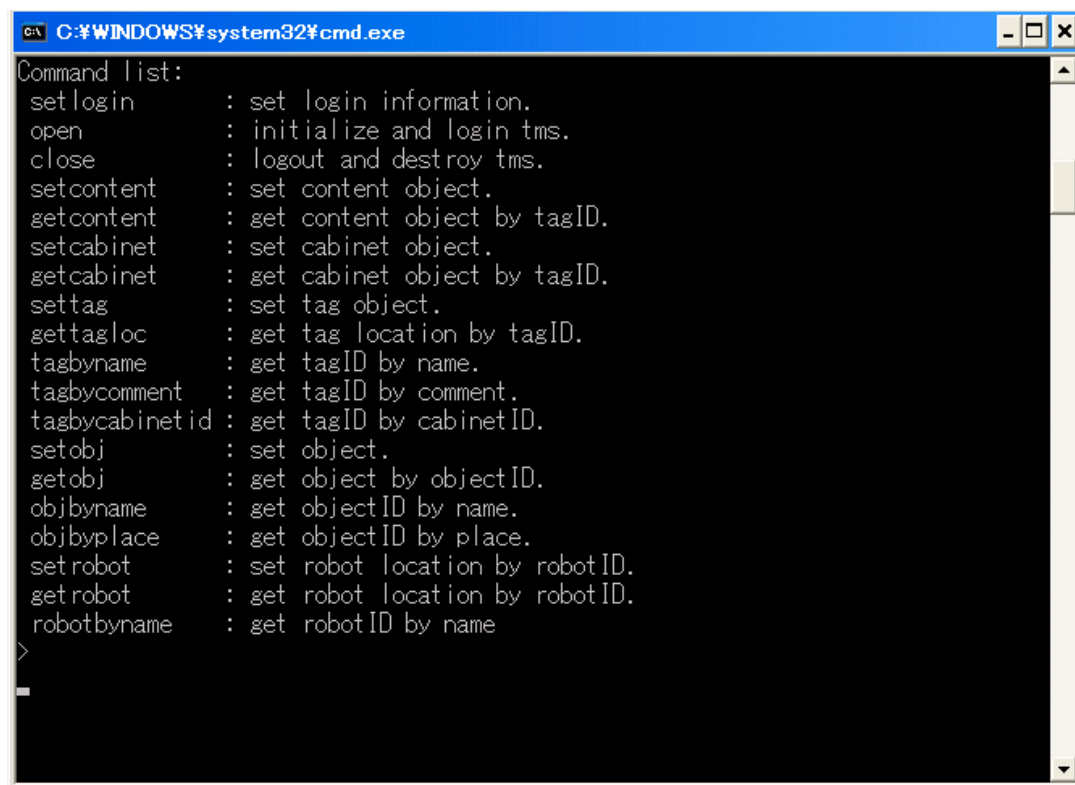


図 1 RTC のリンク

## 4. 7. 実行

活性化後、diConsumer 実行画面より、コマンドを入力し TMS への情報の登録・更新・検索・取得を実行する（図 2）。実行可能なコマンドを表 1 に示す。



```
C:\WINDOWS\system32\cmd.exe
Command list:
setlogin      : set login information.
open          : initialize and login tms.
close         : logout and destroy tms.
setcontent    : set content object.
getcontent    : get content object by tagID.
setcabinet    : set cabinet object.
getcabinet    : get cabinet object by tagID.
settag        : set tag object.
gettagloc     : get tag location by tagID.
tagbyname     : get tagID by name.
tagbycomment  : get tagID by comment.
tagbycabinetid : get tagID by cabinetID.
setobj        : set object.
getobj        : get object by objectID.
objbyname     : get objectID by name.
objbyplace    : get objectID by place.
setrobot      : set robot location by robotID.
getrobot      : get robot location by robotID.
robotbyname   : get robotID by name
>
```

図 2 コマンド入力画面

表 1 diConsumer のコマンド一覧

コマンド	説明
setlogin	ログイン情報を設定
open	TMS の初期化とログイン
close	TMS のログアウトと終了
setContent	収納物情報を登録。最初に登録する収納物数を指定。
getContent	タグ ID を指定し収納物情報を取得
setcabinet	収納棚情報を登録。最初に登録する収納棚数を指定。
getcabinet	タグ ID を指定し収納棚情報を取得
settag	タグ情報を登録。最初に登録するタグ数を指定。
gettagloc	タグ ID を指定しタグの位置情報を取得。 収納物の場合、座標変換を自動で行う。
tagbyname	名前によりタグ ID を検索
tagbycomment	コメントによりタグ ID を検索
tagbycabinetid	収納棚 ID よりタグ ID を検索
setobj	移動体情報を登録。最初に登録する移動体数を指定
getobj	移動体 ID を指定し移動体情報を取得
objbyname	名前により移動体 ID を検索
objbyplace	場所により移動体 ID を検索
setrobot	未実装
getrobot	未実装
robotbyname	未実装

## 4. 8. 終了手順

実行を停止する場合は、RTC の非活性化を行う。RT System Editor のツールバーの “All Deactivate” をクリックすることで非活性化を行う。

## 5. 特記事項

本モジュールをご利用される場合には、以下の記載事項・条件にご同意いただいたものとします。

- ドキュメントに情報を掲載する際には万全を期していますが、それらの情報の正確性または利用者にとっての有用性などについては一切保証いたしません。
- 利用者が本モジュールを利用することにより生じたいかなる損害についても一切責任を負いません。
- 本モジュールの変更、削除等は、原則として利用者への予告無しに行います。また、やむを得ない事由により公開を中断あるいは中止させていただくことがあります。
- 本モジュールの情報の変更、削除、公開の中断、中止により、利用者に生じたいかなる損害についても一切責任を負いません。

### 【連絡先】

東京大学 大学院工学系研究科精密工学専攻 浅間研究室

〒113-8656 東京都文京区本郷 7-3-1

Tel : 03-5841-6486

E-Mail : rtc-contact@robot.t.u-tokyo.ac.jp